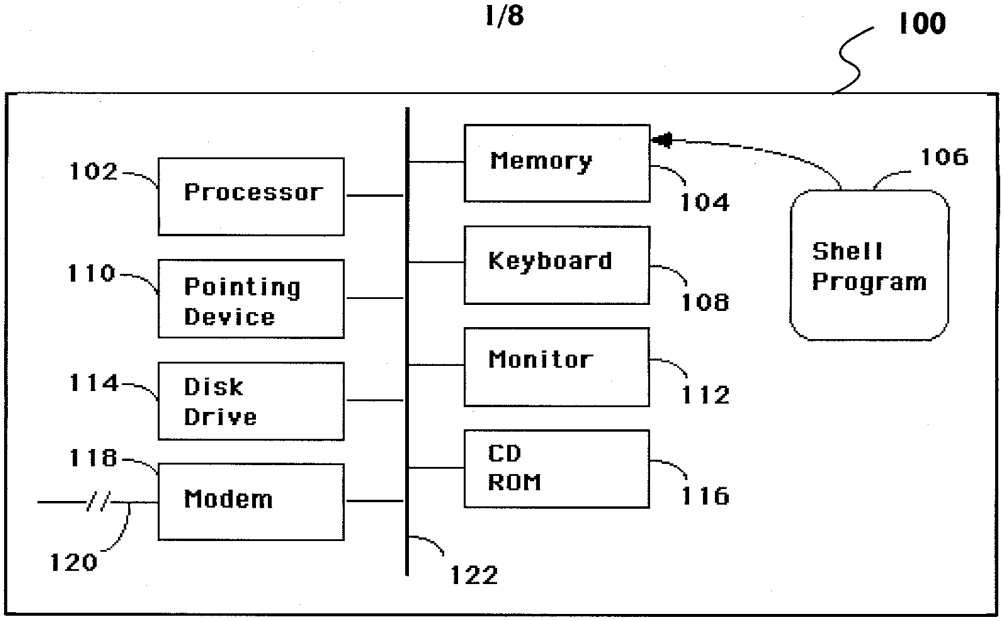

100
102
Processor
110
Pointing Device
114
Disk Drive
118
Modem
120
Memory
104
Keyboard
108
Monitor
112
CD ROM
116
122
106
Shell Program

FIG. 1

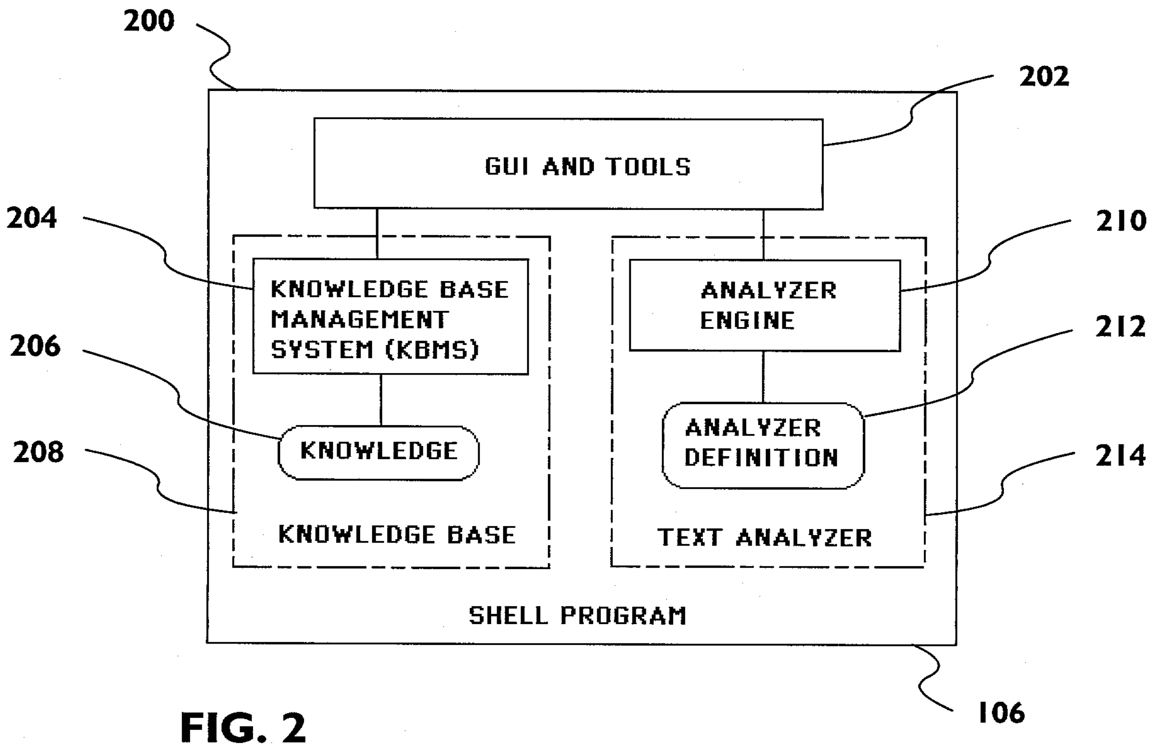
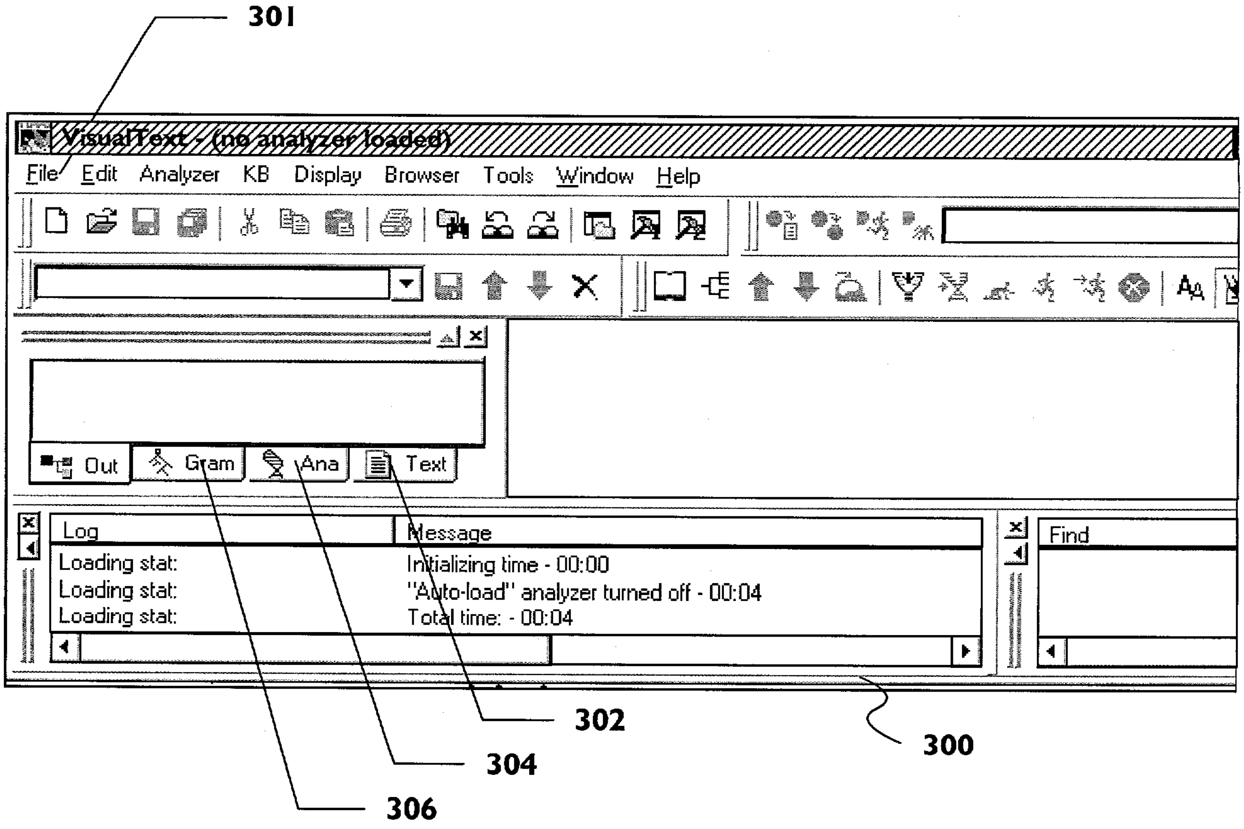
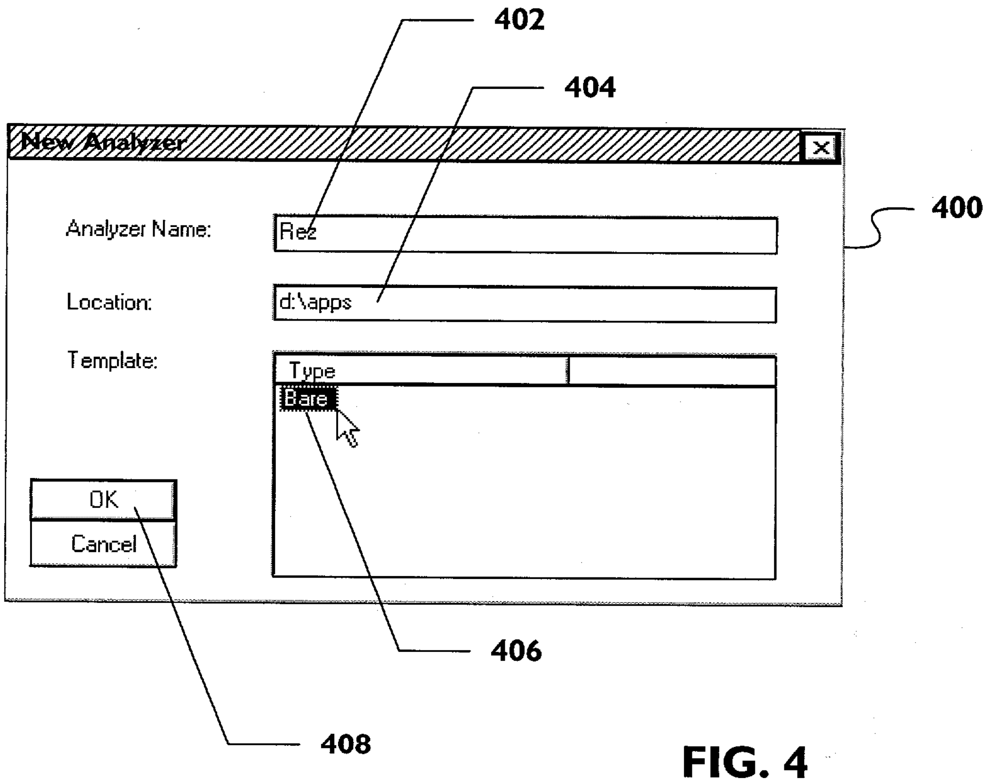

200
202
GUI AND TOOLS
204
KNOWLEDGE BASE MANAGEMENT SYSTEM (KBMS)
206
KNOWLEDGE
208
KNOWLEDGE BASE
210
ANALYZER ENGINE
212
ANALYZER DEFINITION
214
TEXT ANALYZER
SHELL PROGRAM
106

FIG. 2


301
VisualText - (no analyzer loaded)
File Edit Analyzer KB Display Browser Tools Window Help
Out
Gram
Ana
Text
Find
302
304
300
306

Log	Message
Loading stat:	Initializing time - 00:00
Loading stat:	"Auto-load" analyzer turned off - 00:04
Loading stat:	Total time: - 00:04

FIG. 3



Analyzer Name: Rez

Location: d:\apps

Template: Bare

FIG. 4

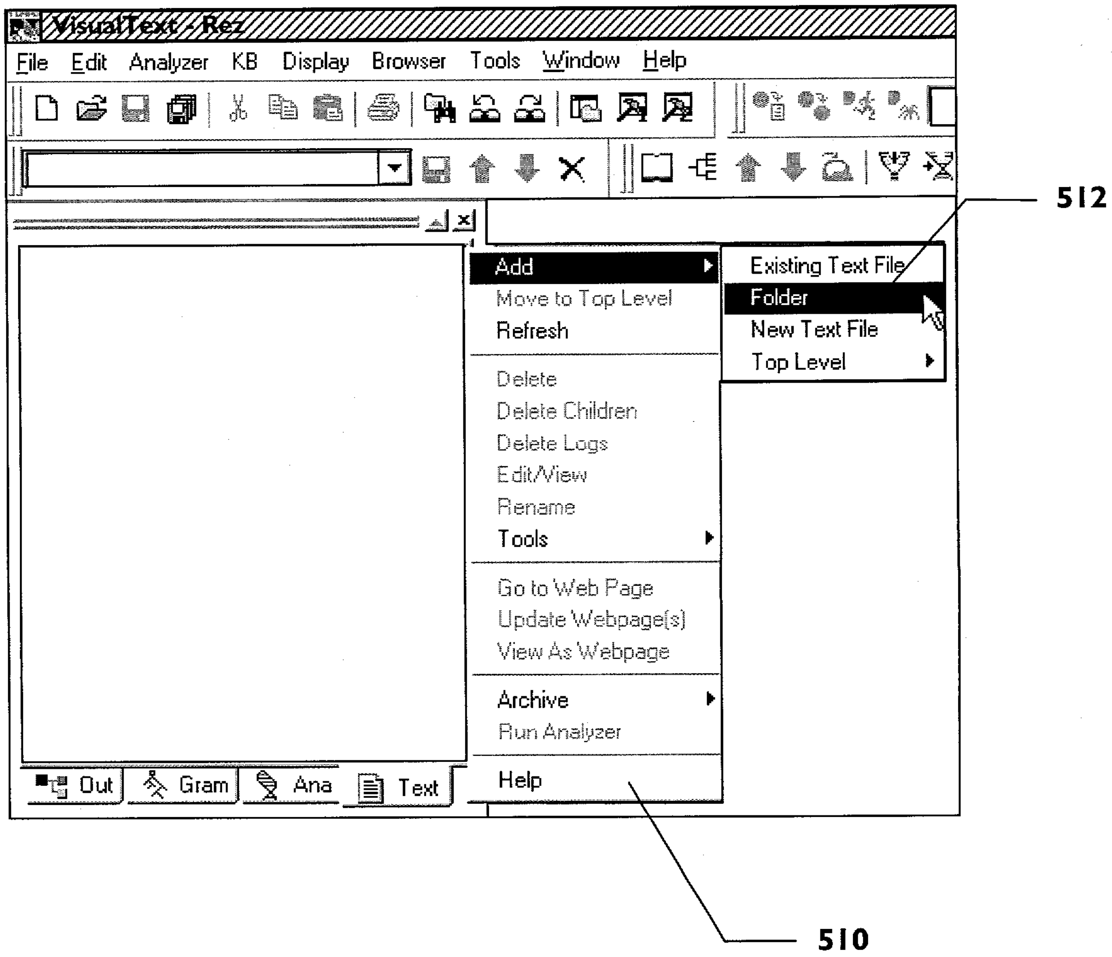
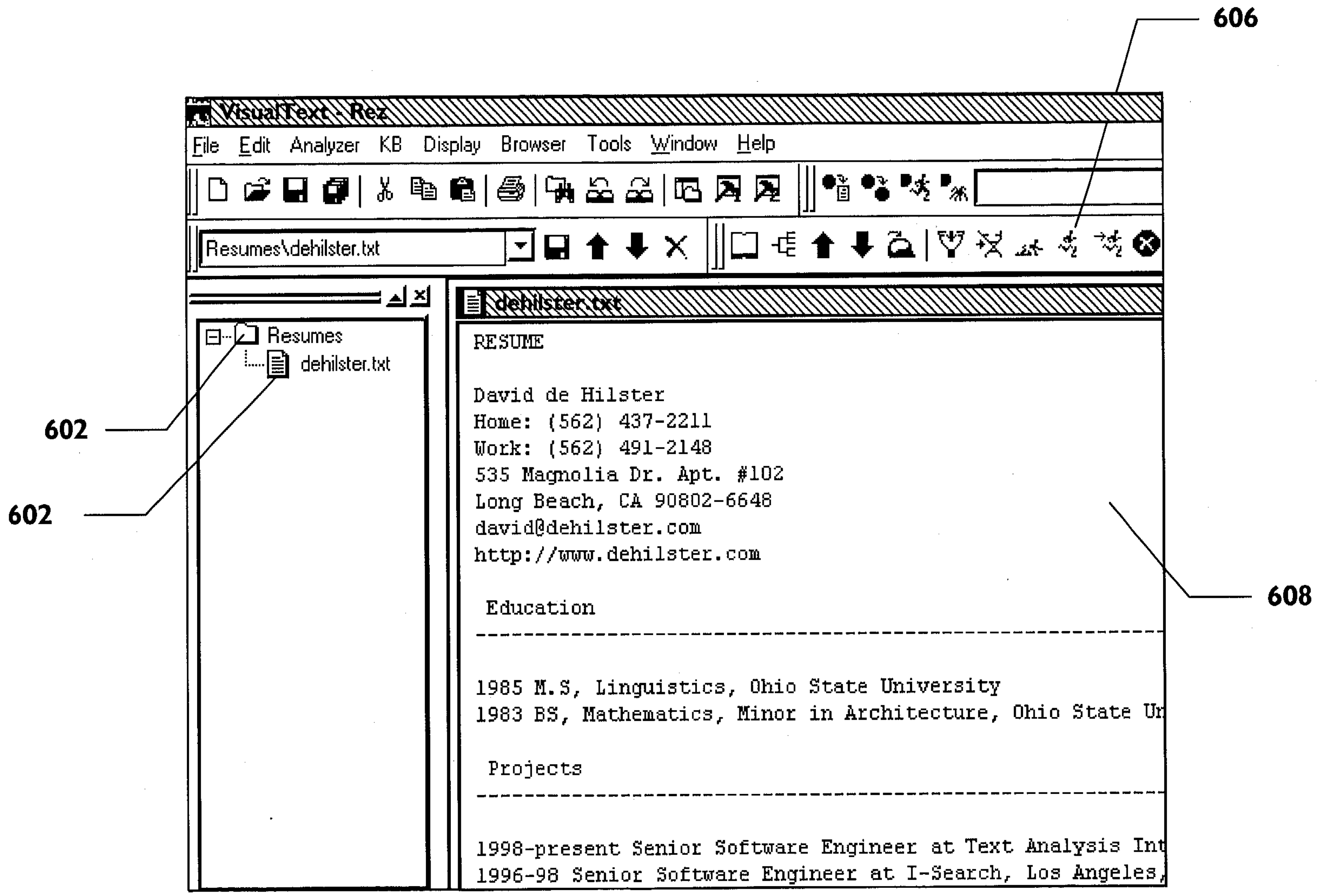
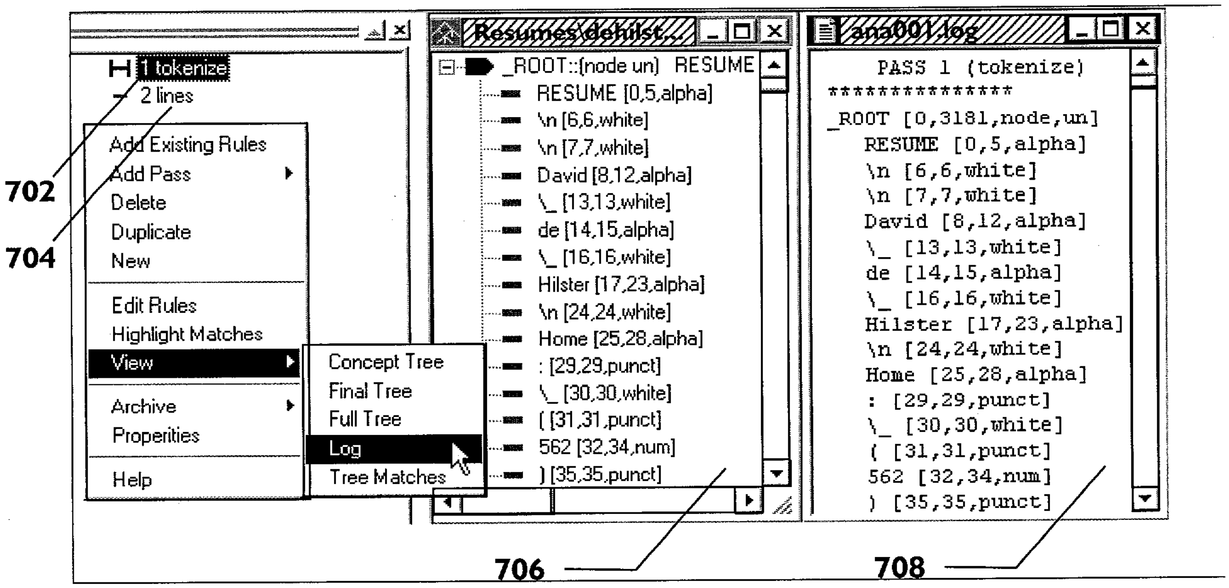

VisualText - Rez
File Edit Analyzer KB Display Browser Tools Window Help
Add
Move to Top Level
Refresh
Delete
Delete Children
Delete Logs
Edit/View
Rename
Tools
Go to Web Page
Update Webpage(s)
View As Webpage
Archive
Run Analyzer
Help
Existing Text File
Folder
New Text File
Top Level
Out
Gram
Ana
Text
512
510

FIG. 5



```
RESUME

David de Hilster
Home: (562) 437-2211
Work: (562) 491-2148
535 Magnolia Dr. Apt. #102
Long Beach, CA 90802-6648
david@dehilster.com
http://www.dehilster.com

 Education
---------------------------------------------------------------

1985 M.S, Linguistics, Ohio State University
1983 BS, Mathematics, Minor in Architecture, Ohio State Ur

 Projects
---------------------------------------------------------------

1998-present Senior Software Engineer at Text Analysis Int
1996-98 Senior Software Engineer at I-Search, Los Angeles,
```

FIG. 6



```
_ROOT::(node un)  RESUME
    RESUME [0,5,alpha]
    \n [6,6,white]
    \n [7,7,white]
    David [8,12,alpha]
    \_ [13,13,white]
    de [14,15,alpha]
    \_ [16,16,white]
    Hilster [17,23,alpha]
    \n [24,24,white]
    Home [25,28,alpha]
    : [29,29,punct]
    \_ [30,30,white]
    ( [31,31,punct]
    562 [32,34,num]
    ) [35,35,punct]
```

```
    PASS 1 (tokenize)
***************
_ROOT [0,3181,node,un]
   RESUME [0,5,alpha]
   \n [6,6,white]
   \n [7,7,white]
   David [8,12,alpha]
   \_ [13,13,white]
   de [14,15,alpha]
   \_ [16,16,white]
   Hilster [17,23,alpha]
   \n [24,24,white]
   Home [25,28,alpha]
   : [29,29,punct]
   \_ [30,30,white]
   ( [31,31,punct]
   562 [32,34,num]
   ) [35,35,punct]
```

FIG. 7

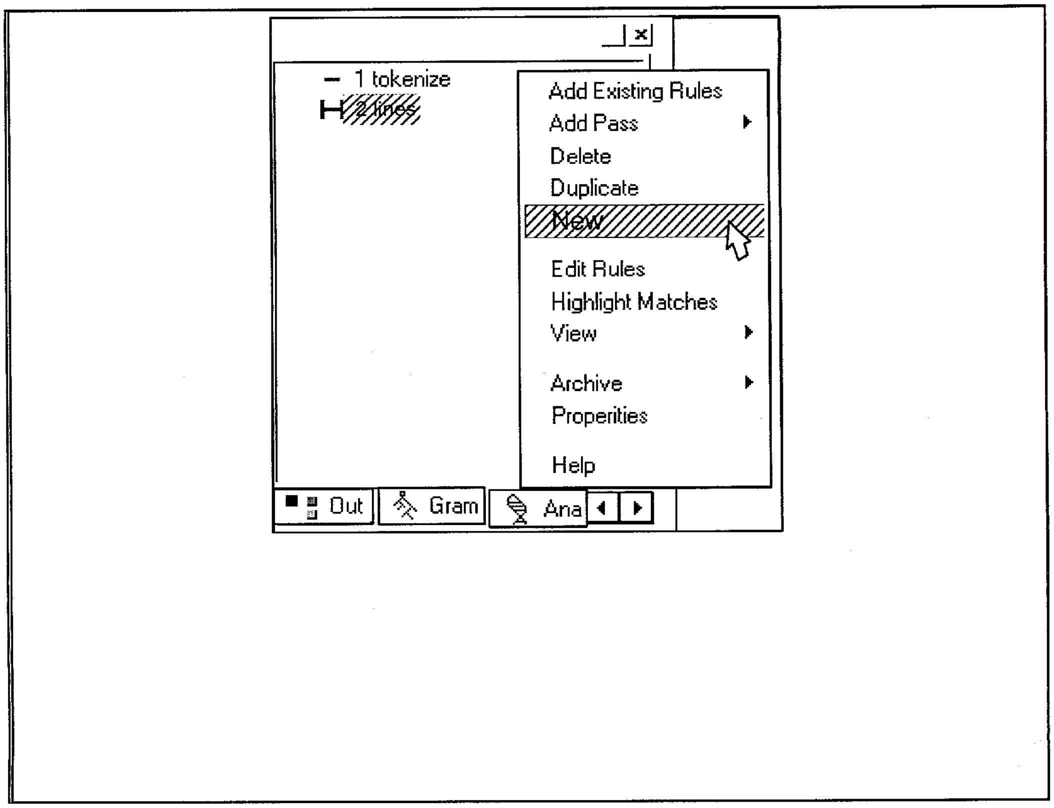
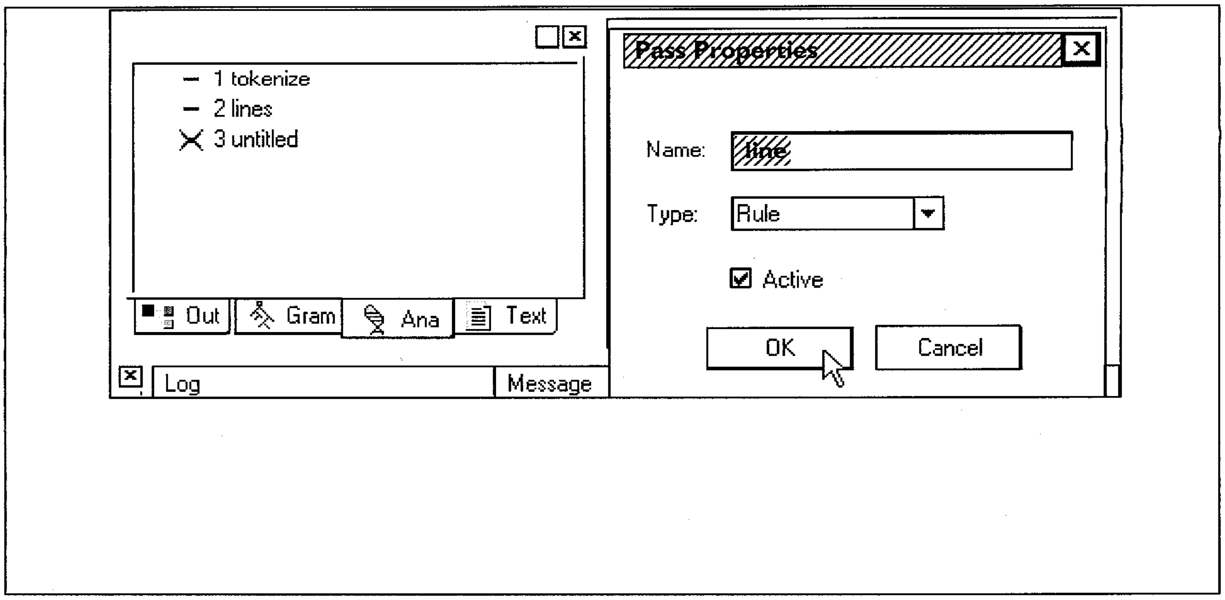


FIG. 8



- 1 tokenize
- 2 lines
- 3 untitled

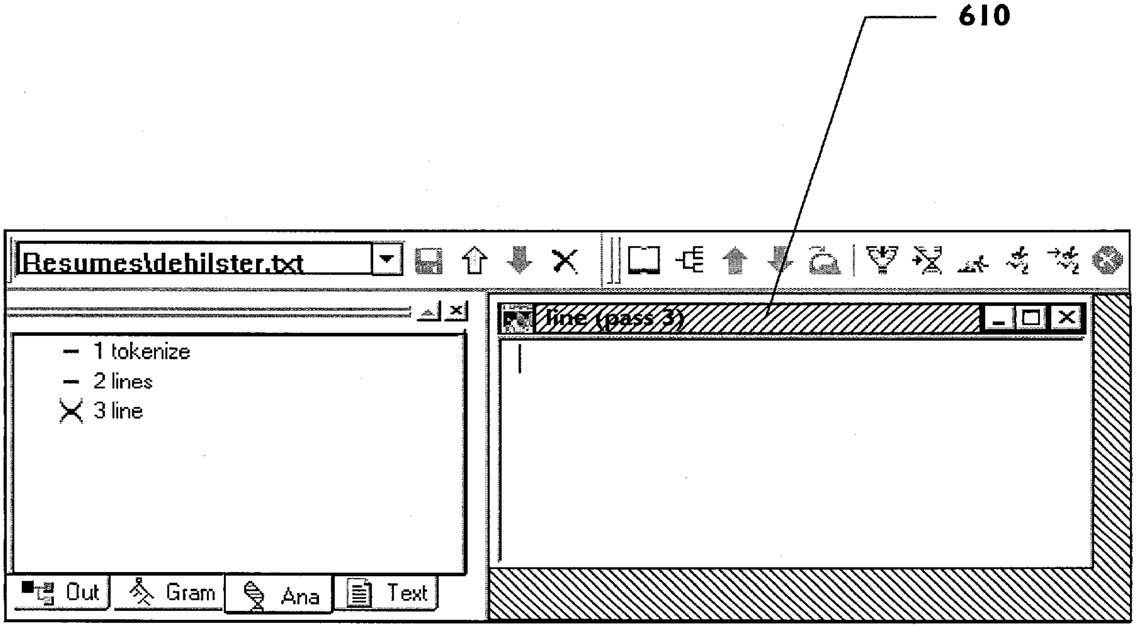
Pass Properties

Name: line

Type: Rule

☑ Active

FIG. 9

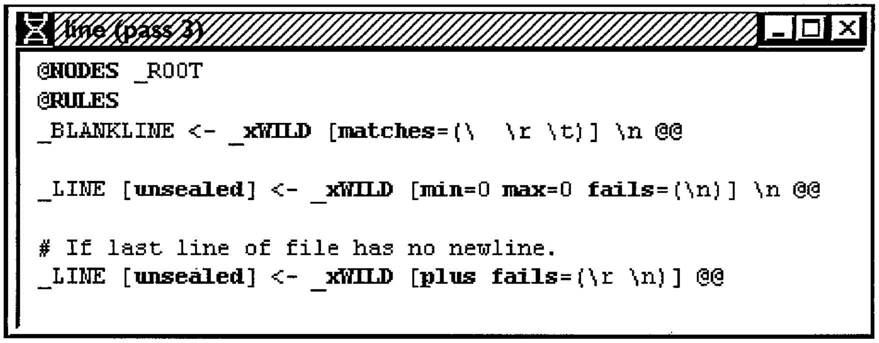


Resumes\dehilster.txt

line (pass 3)

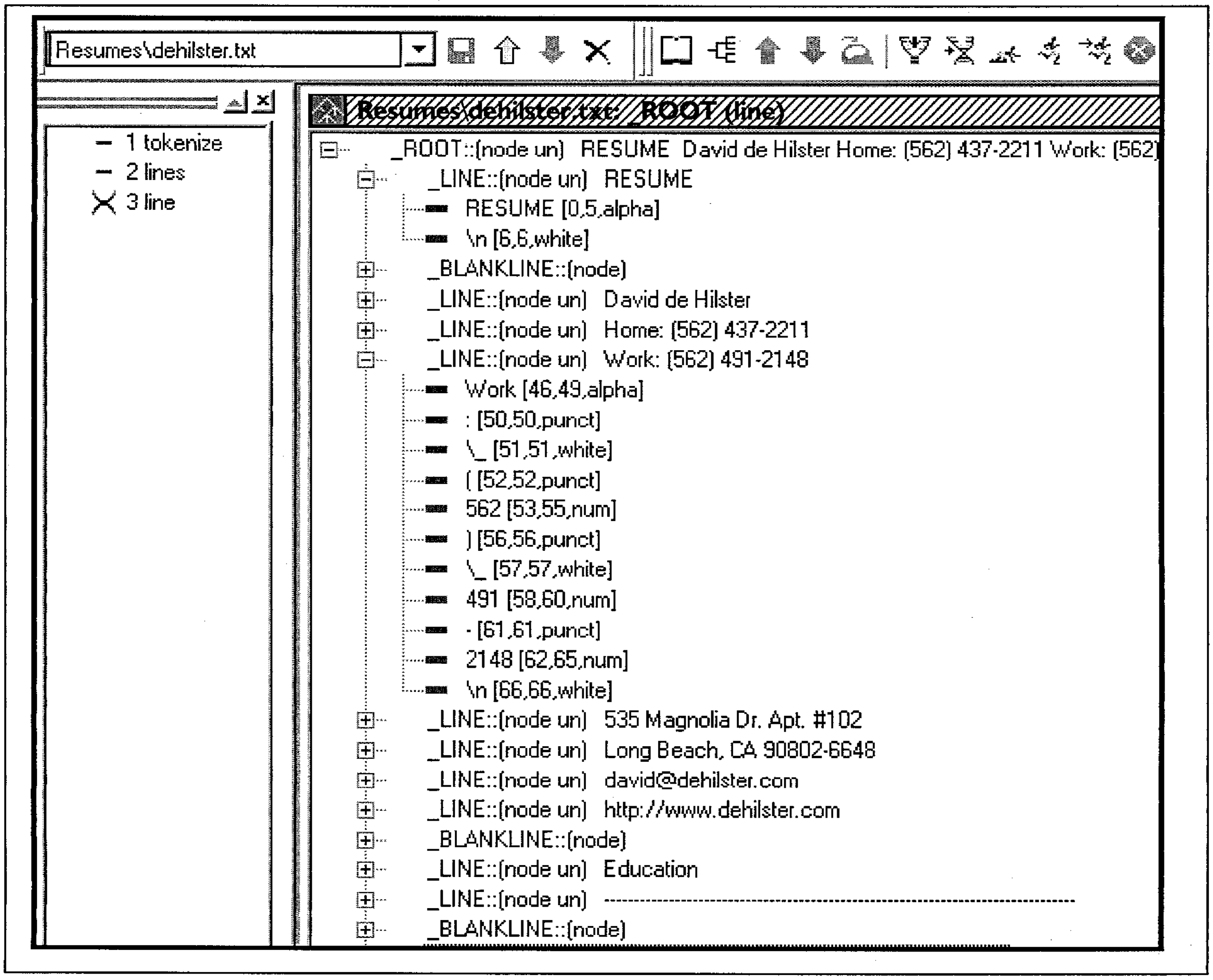
- 1 tokenize
- 2 lines
- 3 line

FIG. 10



```
@NODES _ROOT
@RULES
_BLANKLINE <- _xWILD [matches=(\  \r \t)] \n @@

_LINE [unsealed] <- _xWILD [min=0 max=0 fails=(\n)] \n @@

# If last line of file has no newline.
_LINE [unsealed] <- _xWILD [plus fails=(\r \n)] @@
```

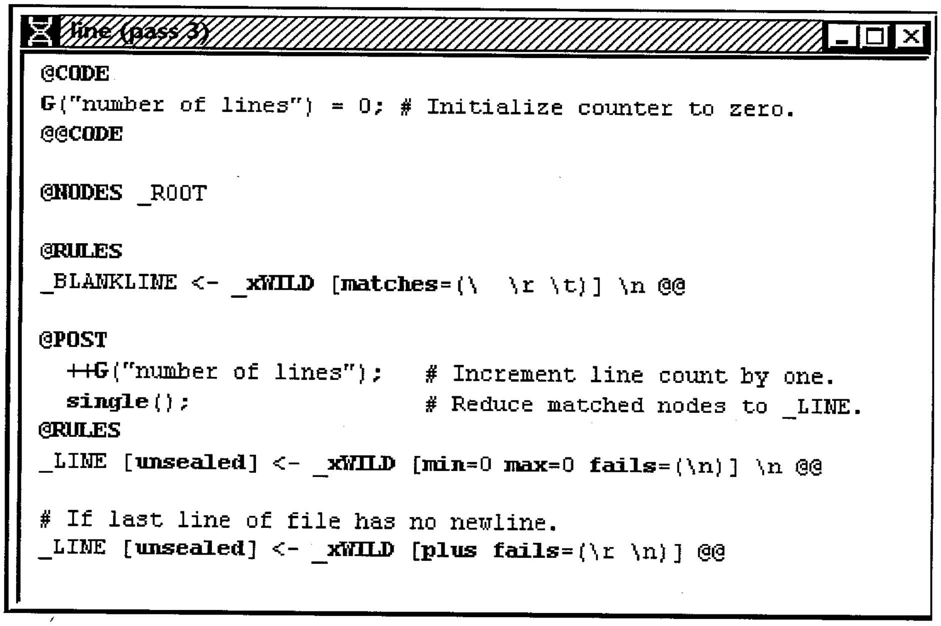
FIG. 11



```
— 1 tokenize
— 2 lines
✕ 3 line
```

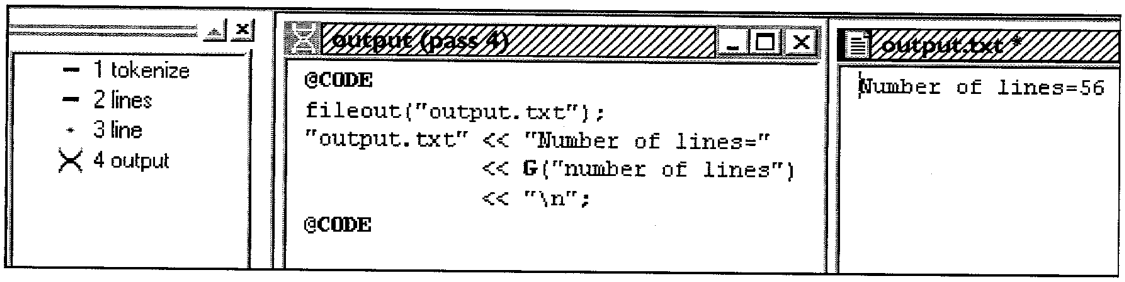
```
⊟ _ROOT::(node un)  RESUME  David de Hilster Home: (562) 437-2211 Work: (562)
   ⊟ _LINE::(node un)  RESUME
        RESUME [0,5,alpha]
        \n [6,6,white]
   ⊞ _BLANKLINE::(node)
   ⊞ _LINE::(node un)  David de Hilster
   ⊞ _LINE::(node un)  Home: (562) 437-2211
   ⊟ _LINE::(node un)  Work: (562) 491-2148
        Work [46,49,alpha]
        : [50,50,punct]
        \_ [51,51,white]
        ( [52,52,punct]
        562 [53,55,num]
        ) [56,56,punct]
        \_ [57,57,white]
        491 [58,60,num]
        - [61,61,punct]
        2148 [62,65,num]
        \n [66,66,white]
   ⊞ _LINE::(node un)  535 Magnolia Dr. Apt. #102
   ⊞ _LINE::(node un)  Long Beach, CA 90802-6648
   ⊞ _LINE::(node un)  david@dehilster.com
   ⊞ _LINE::(node un)  http://www.dehilster.com
   ⊞ _BLANKLINE::(node)
   ⊞ _LINE::(node un)  Education
   ⊞ _LINE::(node un)  ----------------------------------------------------------------
   ⊞ _BLANKLINE::(node)
```

FIG. 12



```
@CODE
G("number of lines") = 0; # Initialize counter to zero.
@@CODE

@NODES _ROOT

@RULES
_BLANKLINE <- _xWILD [matches=(\  \r \t)] \n @@

@POST
  ++G("number of lines");   # Increment line count by one.
  single();                 # Reduce matched nodes to _LINE.
@RULES
_LINE [unsealed] <- _xWILD [min=0 max=0 fails=(\n)] \n @@

# If last line of file has no newline.
_LINE [unsealed] <- _xWILD [plus fails=(\r \n)] @@
```

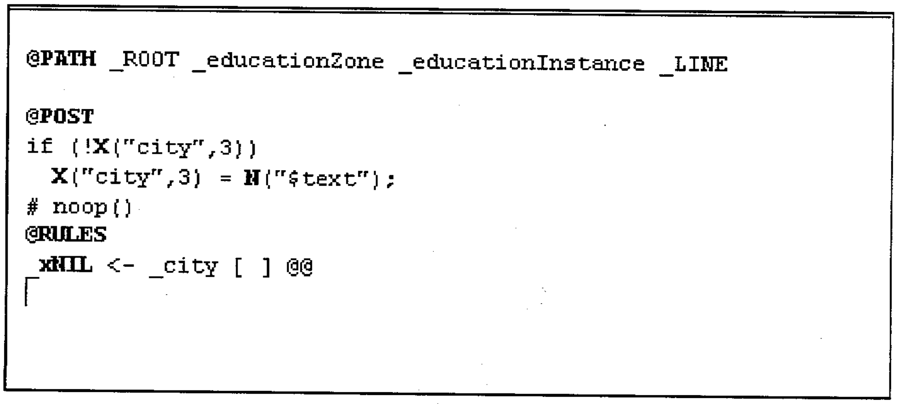
FIG. 13



```
@CODE
fileout("output.txt");
"output.txt" << "Number of lines="
             << G("number of lines")
             << "\n";
@CODE
```

```
Number of lines=56
```

FIG. 14



```
@PATH _ROOT _educationZone _educationInstance _LINE

@POST
if (!X("city",3))
  X("city",3) = N("$text");
# noop()
@RULES
_xNIL <- _city [ ] @@
```

FIG. 15